

Лабораторная работа № 2 по курсу : Операционные системы

Выполнил студент группы М8О-201Б-21 *Кварацхелия Александр.*

Условие

Ознакомиться с сигналами операционной системы UNIX/LINUX, используя утилиту strace, проанализировать результаты, сопоставить их с кодом программы.

Метод решения

Использовать свободно распространяемую утилиту strace следующим образом:
strace lab2

Вывод strace

```
execve("./../lab2/child", ["./../lab2/child"], 0x7fff5240b618 /* 52 var  
brk(NULL) = 0x55801db00000  
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3  
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=153051, ...}, AT_EMPTY_PATH)  
mmap(NULL, 153051, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe68d0f6000  
close(3) = 0  
openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3  
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P4\2\0\0\0\0\0"... , 83  
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...  
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1953472, ...}, AT_EMPTY_PATH)  
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x  
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...  
mmap(NULL, 1994384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe68cf0d  
mmap(0x7fe68cf2f000, 1421312, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_  
mmap(0x7fe68d08a000, 356352, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,  
mmap(0x7fe68d0e1000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_  
mmap(0x7fe68d0e7000, 52880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_  
close(3) = 0  
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x  
arch_prctl(ARCH_SET_FS, 0x7fe68d0f5640) = 0  
set_tid_address(0x7fe68d0f5910) = 22440  
set_robust_list(0x7fe68d0f5920, 24) = 0  
rseq(0x7fe68d0f5f60, 0x20, 0, 0x53053053) = 0  
mprotect(0x7fe68d0e1000, 16384, PROT_READ) = 0
```

```

mprotect(0x55801d6b3000, 4096, PROT_READ) = 0
mprotect(0x7fe68d14d000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}, ...) = 0
munmap(0x7fe68d0f6000, 153051) = 0
getrandom("\xb4\x8b\x18\xb4\x43\x6a\xe9\x9b", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55801db00000
brk(0x55801db21000) = 0x55801db21000
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EACCESS) = 0
read(0, somestring
"somestring\n", 1024) = 11
write(1, "smstrng\n", 8smstrng
) = 8
read(0, shortstring545
"shortstring545\n", 1024) = 15
write(1, "shrtstrng545\n", 13shrtstrng545
) = 13
read(0, "", 1024) = 0
write(1, "", 0) = 0
exit_group(0) = ?
+++ exited with 0 +++

```

Выводы

Вызов *fork* дублирует породивший его процесс со всеми его переменными, файловыми дескрипторами, приоритетами процесса, рабочий и корневой каталоги, и сегментами выделенной памяти.

Ребёнок **не** наследует:

- идентификатора процесса (PID, PPID);
- израсходованного времени ЦП (оно обнуляется);
- сигналов процесса-родителя, требующих ответа;
- заблокированных файлов (record locking).

В процессе выполнения лабораторной работы были приобретены навыки практического применения создания, обработки и отслеживания их состояния. Для выполнения данного варианта задания создание потоков как таковых не требуется, так как всю работу выполняет системный вызов «exes».